Setting Up & Running the WRF Standard Initialization

Paula McCaslin

John Smart, Brent Shaw and Matt Pyle

NOAA / Forecast Systems Laboratory

Sept 27, 2005

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Overview

The WRF modeling system includes 2 dynamic cores:

- Advanced Research WRF (ARW) developed by NCAR/MMM (formerly referred to as the Eulerian Mass core)
- Nonhydrostatic Mesoscale Model (NMM) developed by NOAA/NCEP

Each dynamic core currently has a separate SI package and graphical user interface (GUI)

- Options unique to each package are noted in this presentation

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

SI Capabilities

- To provide 3 mandatory functions to prepare data for WRF simulations:

1. Define mesoscale simulation domains (and nests ARW)
2. Create non-time-varying (static) terrain and land state variables files for the domain grids (land use, soil type, etc)
3. Provide initial and lateral boundary condition files for “real” data cases on these domains
    - De-GRIB files to acquire time-varying meteorological data (soil layers, sea-surface temp, etc)
    - Interpolate meteorological data to WRF grid -- horizontally and vertically, including grid staggering

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Source Code

- SI Source code
  - Available at http:/wrfsi.noaa.gov/release
  - ARW v2.1.1 - 18 July 2005 (wrfsi_v2.1.1.tar.gz 22 MB)
  - NMM v2.1.1 - 27 Sep 2005 (wrfnmm_si.tar.gz 22 MB)
- SI non-time-varying data files
  - Geographical and surface characteristics data files (6 GB)
  - Available at http:/wrfsi.noaa.gov/release, or ftp://aftp.fsl.noaa.gov/divisions/frd-laps/WRFSI/Geog_Data

Source Code

- Source code data files – Static geographical and surface characteristics data
  - Topo – grid average elevation:
    - *topo_30s/topo_30s_[NW, NE, SW, SE].tar.gz*
  - Land use dominant categories (wetland, water, forest):
    - *landuse_30s/landuse_30s_[NW, NE, SW, SE].tar.gz*
  - Annual greenness fraction (min and max): *greenfrac.tar.gz*
  - Soil temperature, adjusted mean annual:
    - *soiltemp_1deg/T90S000E and T90S180W* (two 180 tiles)
  - Soil type – top layer dominant categories (silt, sand, clay, bedrock):
    - *soiltype_top_30s/soiltype_top_30s.[NW, NE, SW, SE].tar.gz*
  - Soil type – bottom layer dominant categories:
    - *soiltype_bot_30s/soiltype_bot_30s.[NW, NE, SW, SE].tar.gz*
  - Albedo: *albedo_ncep.tar.gz*
  - Max Snow Albedo: *maxsnowalb.tar.gz*
  - Terrain slope index: *islope.tar.gz*

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

System Requirements

- Unix or Linux operating system
  - ARW
    - Routinely built on IBM AIX, Intel-Linux, Alpha-Linux at FSL
    - Built on Alpha-True64 and SGI-IRIX at NCAR
  - NMM
    - IBM AIX, Intel-Linux
- FORTRAN 90/95 Compiler
- C Compiler (gcc is preferred)
- NetCDF Libaries
- Perl
- make Utility
- NCAR Graphics with NCAR Graphics Command Language (NCL) - optional

System Requirements

- Disk Requirements
  - Recommended min total space: 10 GB
  - 47 MB: 35 MB for source code, 12 MB executables
  - 6 GB for global static fields
  - 1-10 GB for each domain (in directory MOAD_DATAROOT)
    - Depending on domain size, run length, etc.
    - E.g. NCAR 30km national domain requires approx. 1 GB to contain necessary files for a 48-hour forecast period with 3-hourly boundary conditions
  - 2-4 GB for *typical* initial and lateral boundary data (in directory EXT_DATAROOT)
    - E.g. 48 hours from GFS 1 deg and from ETA 40 km grids with hourly output

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Installing the Software

- Installation Overview

1. Choose directory locations and set environment variables
2. Check compiler options available on your system
3. Check for or build the NetCDF libraries
4. Run the installation script
5. Check for successful installation

Installing the Software

1. Choose directories and set environment variables

The flexibility of the SI allows for a variety of directory location preferences.

Scenario:

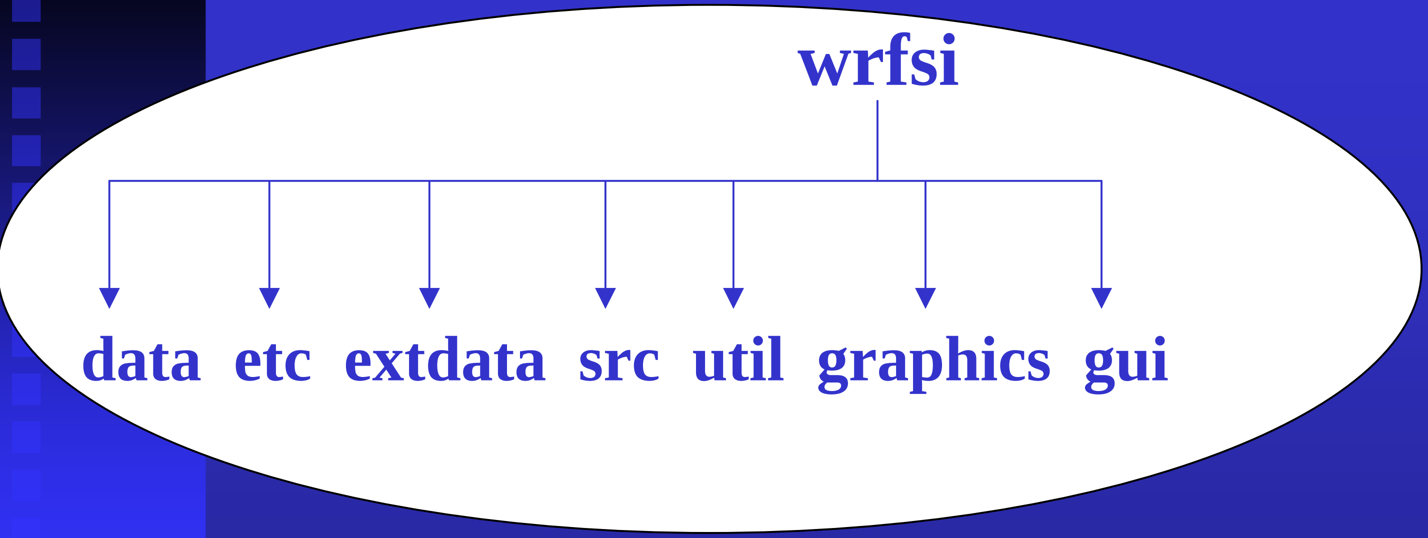
- You prefer to not have executables co-located with source code.
- You prefer to write model ready data somewhere other than with the source code.
- You have the desire to write sizable intermediate data files to a scratch partition.
- Your office has multiple users wanting to use SI, so you probably only need one copy of the 6 GB geography data.

Installing the Software

1. Choose directories and set environment variables

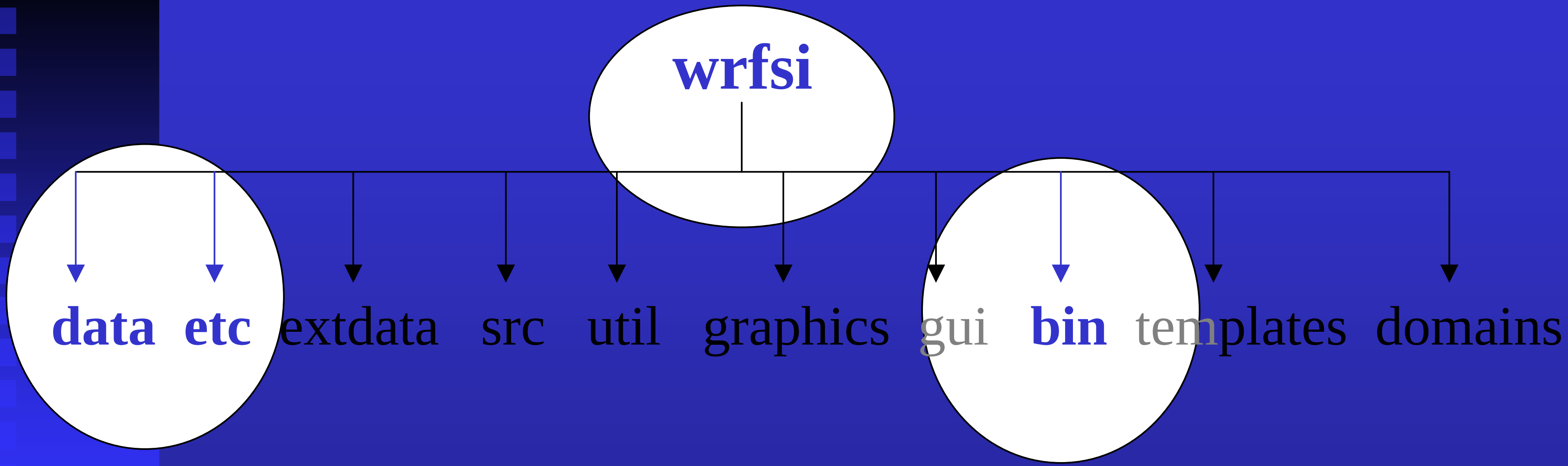
- By default, the SI suite of directory paths will be co-located within the wrfsi SOURCE_ROOT directory.

- Any number of the (seven) SI directory paths can be located *anywhere* on your system, if you simply define the env variables prior to the installation process.



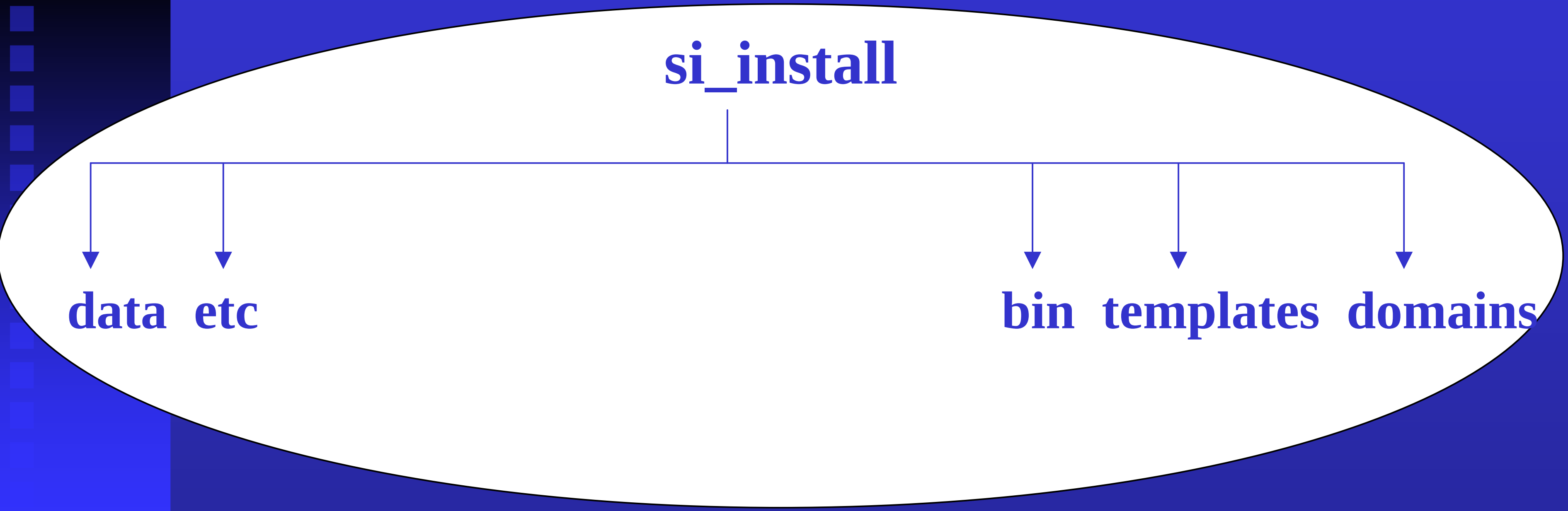
SOURCE_ROOT

SI directory structure when code is first extracted from the SI tape archive (tar) file, typically a subdirectory of WRF. This top level dir is called wrfsi_nmm, for the NMM.



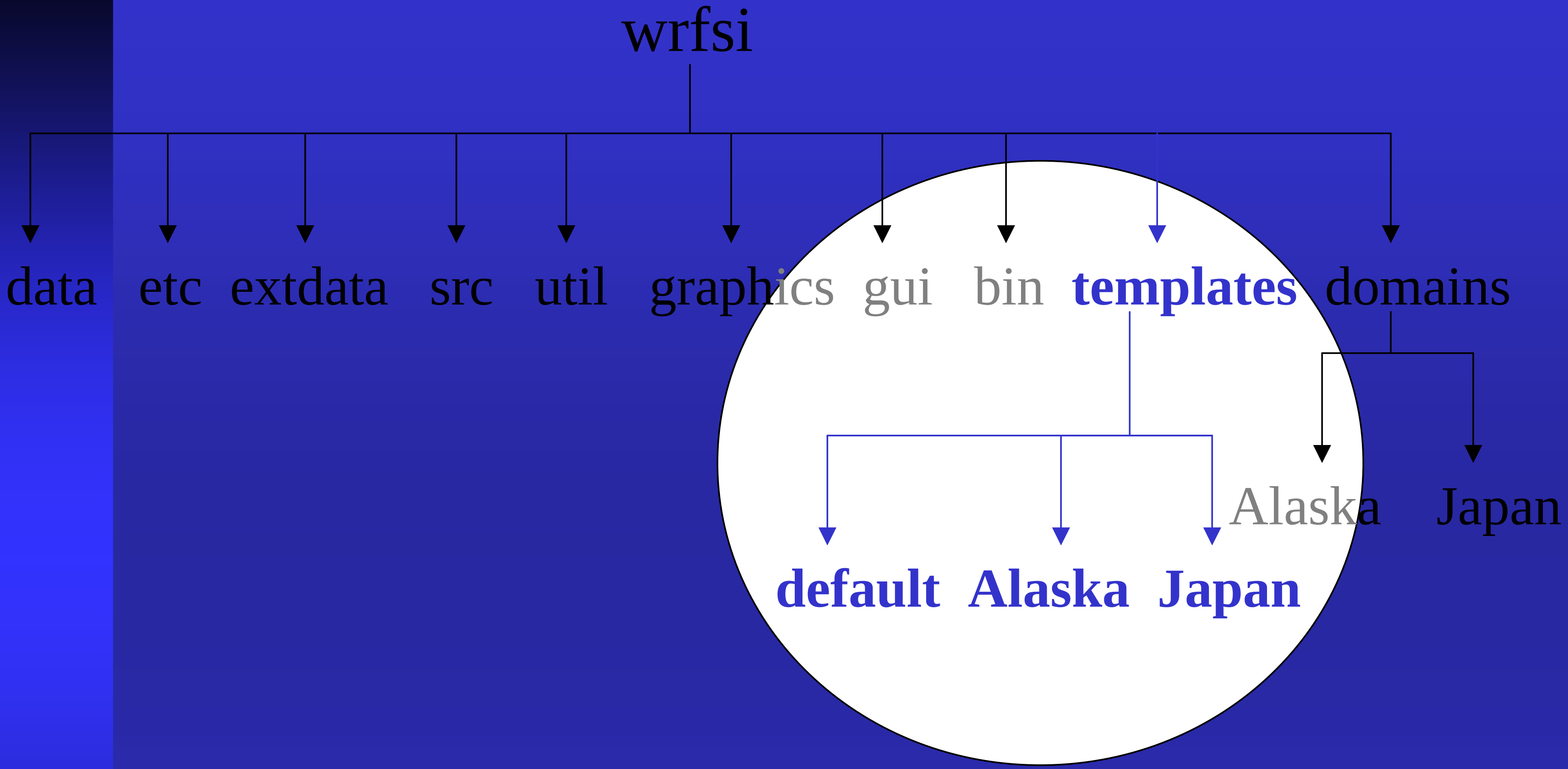
INSTALLROOT

SI directory structure where the compiled binary executables and scripts will be found after installation process (running install_wrfsi.pl). Without setting a value for the INSTALLROOT path, it is co-located in SOURCE_ROOT.



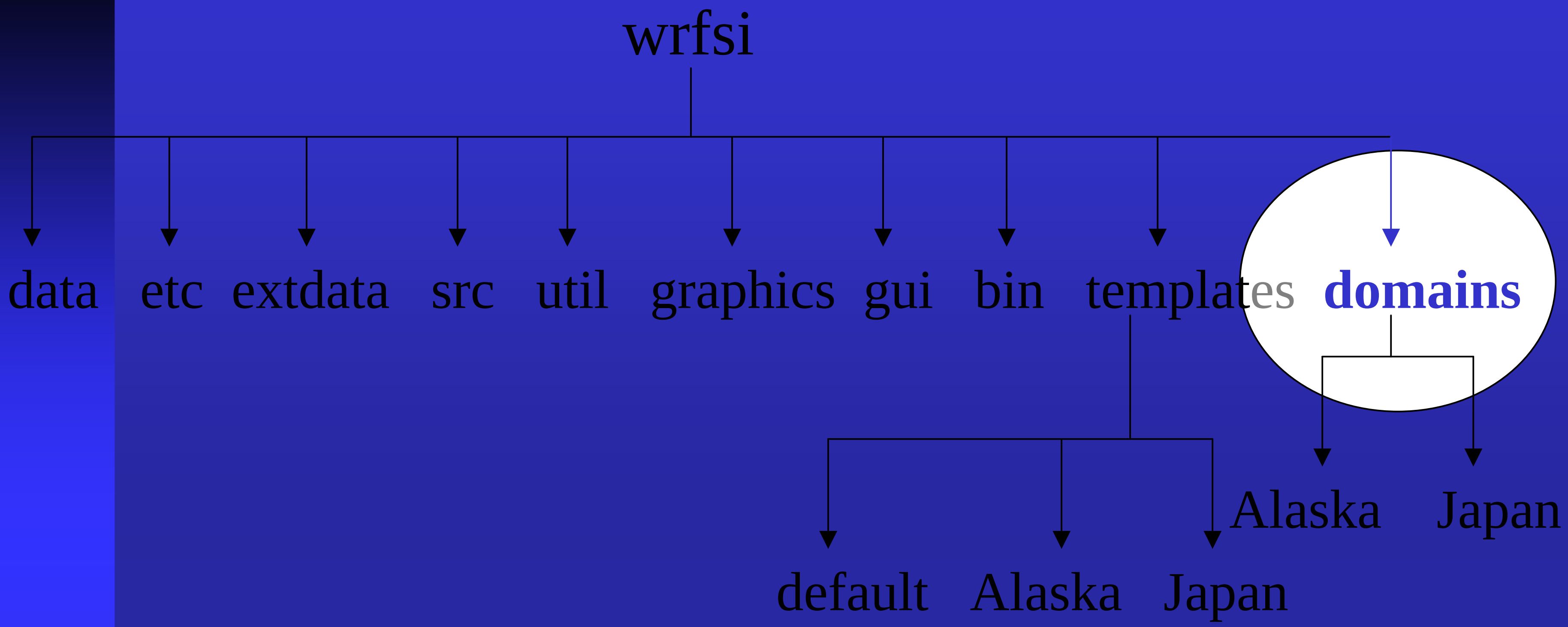
INSTALLROOT

SI directory structure after installation processes where INSTALLROOT is set to a directory location other than SOURCE_ROOT. A user might want, for example, two OS builds from a common SOURCE_ROOT named si_aix and si_linux.



TEMPLATES

Default directory location for subdirectories that will contain a domain’s template directory and namelist, wrfsi.nl.



wrfsi

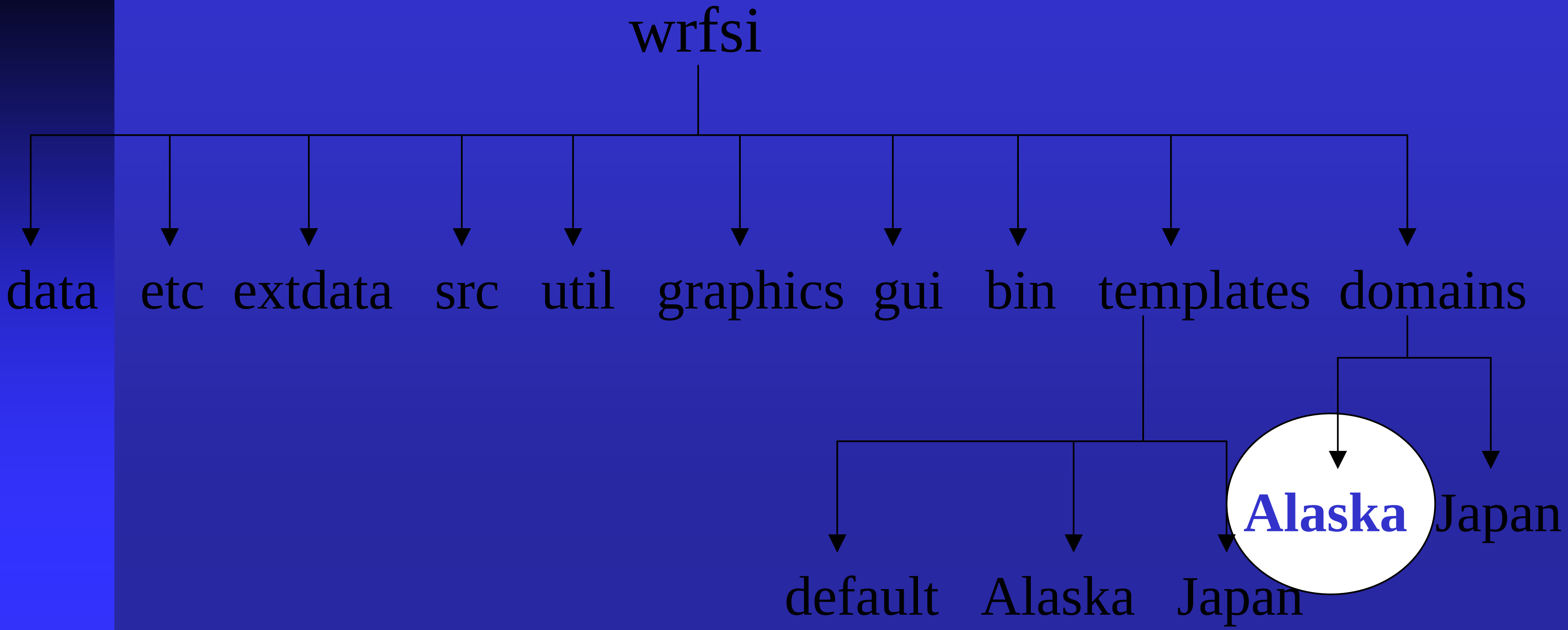
data etc extdata src util graphics gui bin templates domains

default Alaska Japan

Alaska Japan

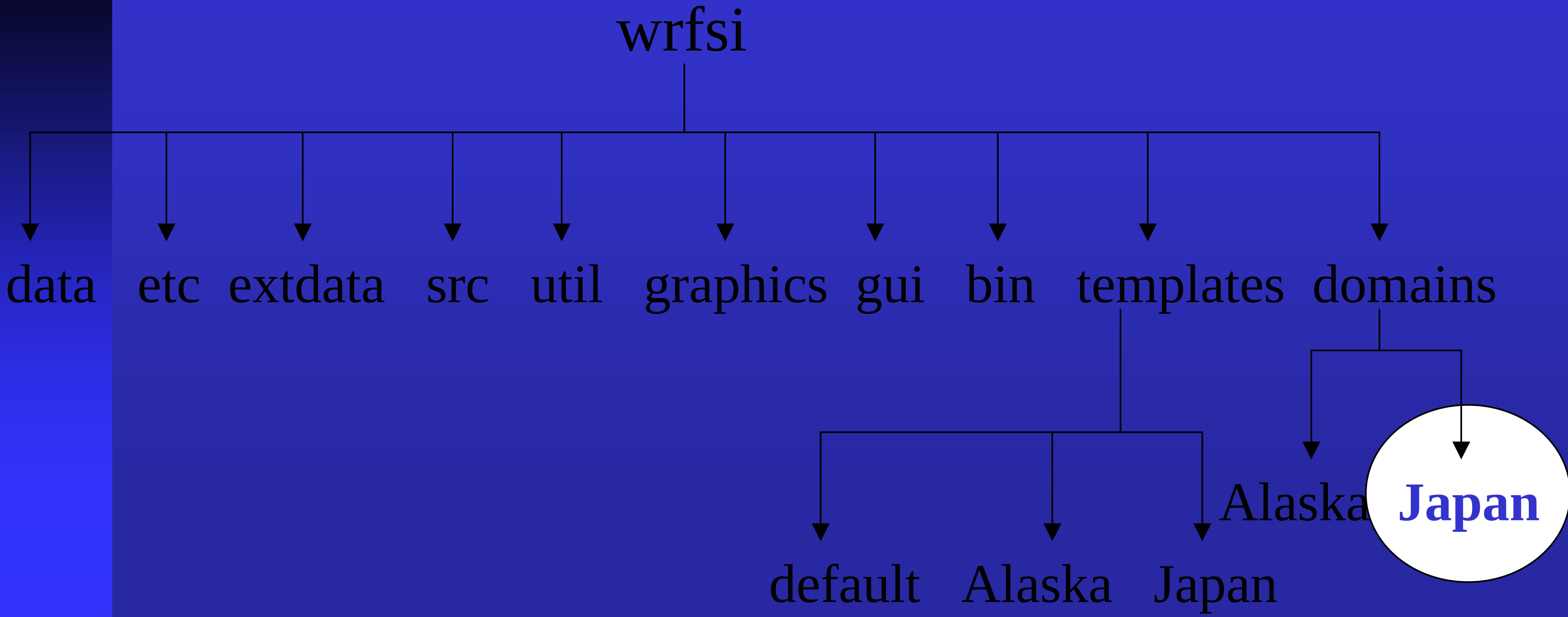
DATAROOT

Directory path where the domain MOAD_DATAROOTs will be located. (Often located in dir other than SOURCE_ROOT in order to preserve previously created domains, esp when upgrading new SI source code.)



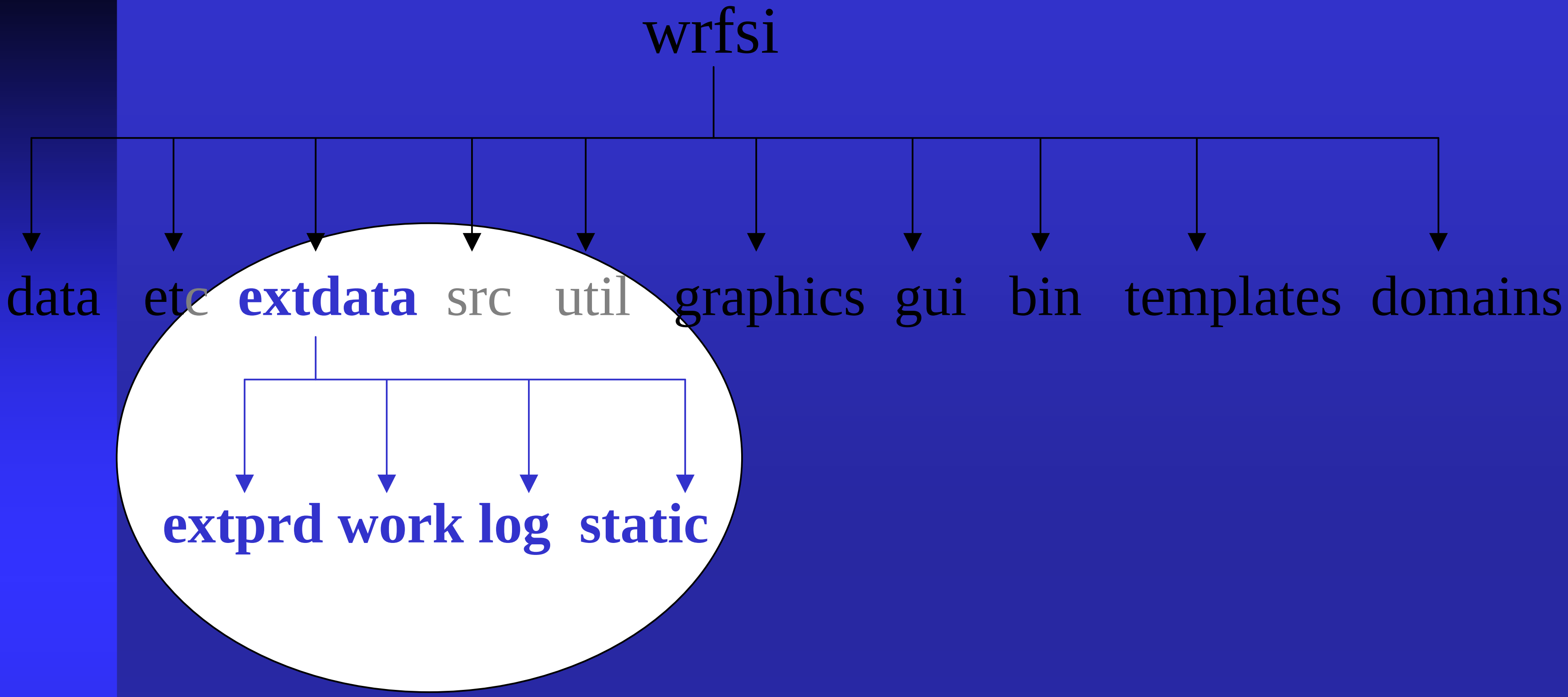
MOAD_DATAROOT

Directory where a domain's definition files and the wrf input data will be written.



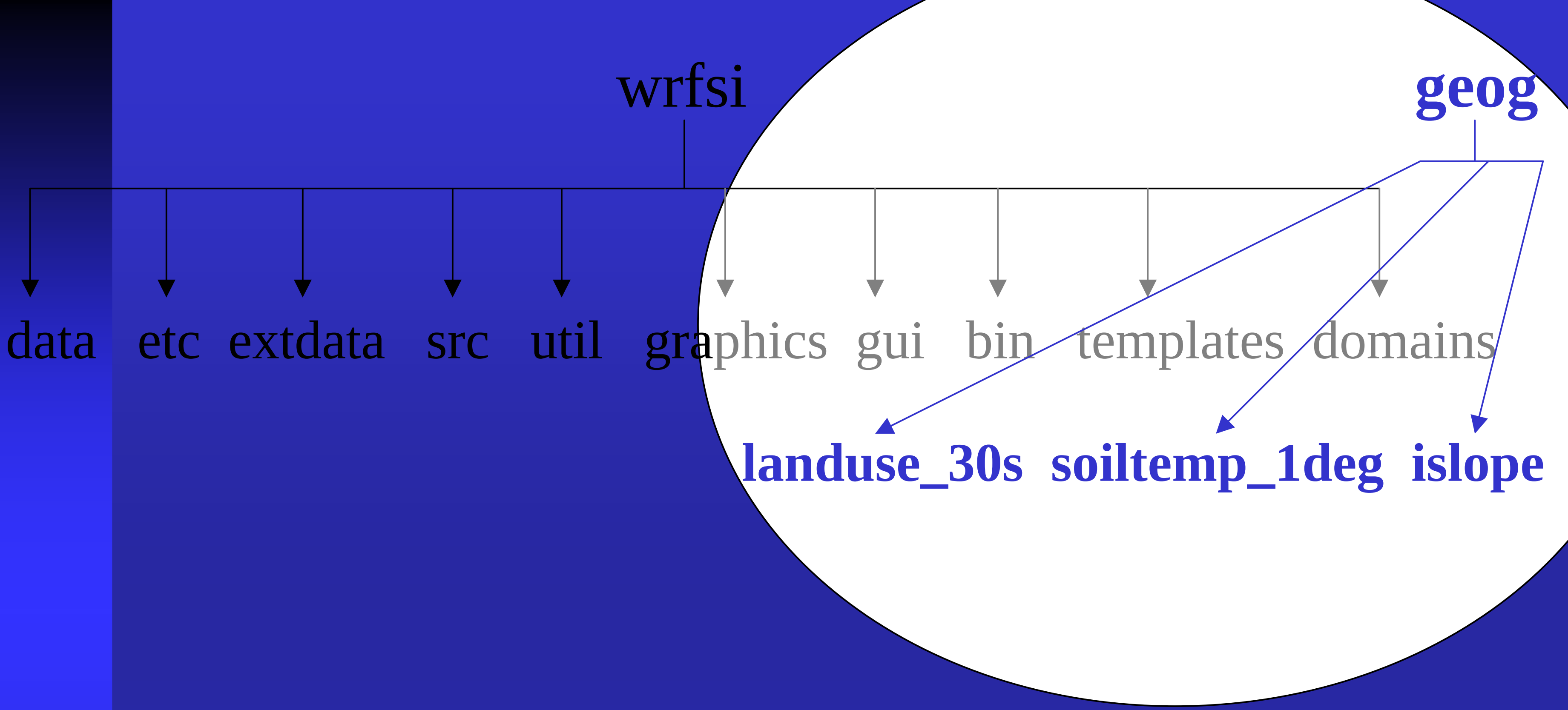
MOAD_DATAROOT

Example of multiple MOAD_DATAROOTs.



EXT_DATAROOT

Directory path containing output from grib_prep process (the decoded and time interpolated GRIB files). This dir supports multiple MOAD_DATAROOTs. Allow at least 2GB. The subdirectories are created first time install_wrfsi.pl is run.



GEOG_DATAROOT

Directory path containing geography data subdirectories, landuse_30s, etc.

Installing the Software

1. Choose directories and set environment variables (con't)

   - Example directory paths configured as environment vars (in csh)

     ```
     setenv SOURCE_ROOT /home/WRFV2/wrfsi_nmm
     setenv INSTALLROOT /home/WRFV2/si_linux
     setenv TEMPLATES /data/wrf/templates
     setenv DATAROOT /data/wrf/domains
     setenv MOAD_DATAROOT /data/wrf/domains/Alaska
     setenv GEOG_DATAROOT /data/geog
     ```

Installing the Software

2. Check Compiler Settings

- Look in SOURCE_ROOT/src/include for the presence of a file named "makefile_{MACH}.inc.in where MACH is the type of machine on which you are building

- There are up to 13 machine specific makefiles for building SI
  - E.g. makefile_ibm.inc.in,
  - makefile_pc.inc.in

- Find and check the makefile’s various compiler options and flag settings for your machine and OS

- If a makefile does not exist for your machine, then copy one of the existing ones to a new name using your machine name then edit its compiler settings to your liking

Installing the Software

3. Check or build the NetCDF libraries and locate Perl

   - Locate path to NetCDF
   - NetCDF must be built with the same compiler used to build SI.
     - E.g. if using pgf90 for SI, then NetCDF must be built with the same version of pgf90.
   - Set environment var **NETCDF** = “Directory path to NetCDF”

   - Locate your systems’ Perl executable (e.g. /usr/bin/perl)
   - Set environment var **PATH_TO_PERL** = “Directory path to Perl”

Installing the Software

4. Run the installation script, install_wrfsi.pl
   - cd $SOURCE_ROOT then run install_wrfsi.pl in one of 3 ways:
   - A) Run install_wrfsi.pl. Let script automatically configure all the directory paths.
   - B) Run install_wrfsi.pl after you have set your SI environment variable directory paths. This is easiest approach and highly recommended for all users.
   - C) Or, run install_wrfsi.pl with command line options to configure directory paths:

```
perl install_wrfsi.pl --installroot=/home/WRFV2/si_aix
--geog_dataroot=/wrf/geog
--path_to_netcdf=/usr/local/netcdf
--machine=ibm
```

(where machine type corresponds to the makefile_{MACH}.inc.in file)

Installing the Software

4. Run the installation script, install_wrfsi.pl (con't)

- When allowing script install_wrfsi.pl to define your environment (option A) would result in the following directory paths:
    - SOURCE_ROOT one dir level above install_wrfsi.pl
    - INSTALLROOT=SOURCE_ROOT
    - TEMPLATES=INSTALLROOT/templates
    - DATAROOT=INSTALLROOT/domains
    - EXT_DATAROOT=INSTALLROOT/extdata
    - GEOG_DATAROOT=EXT_DATAROOT/GEOG

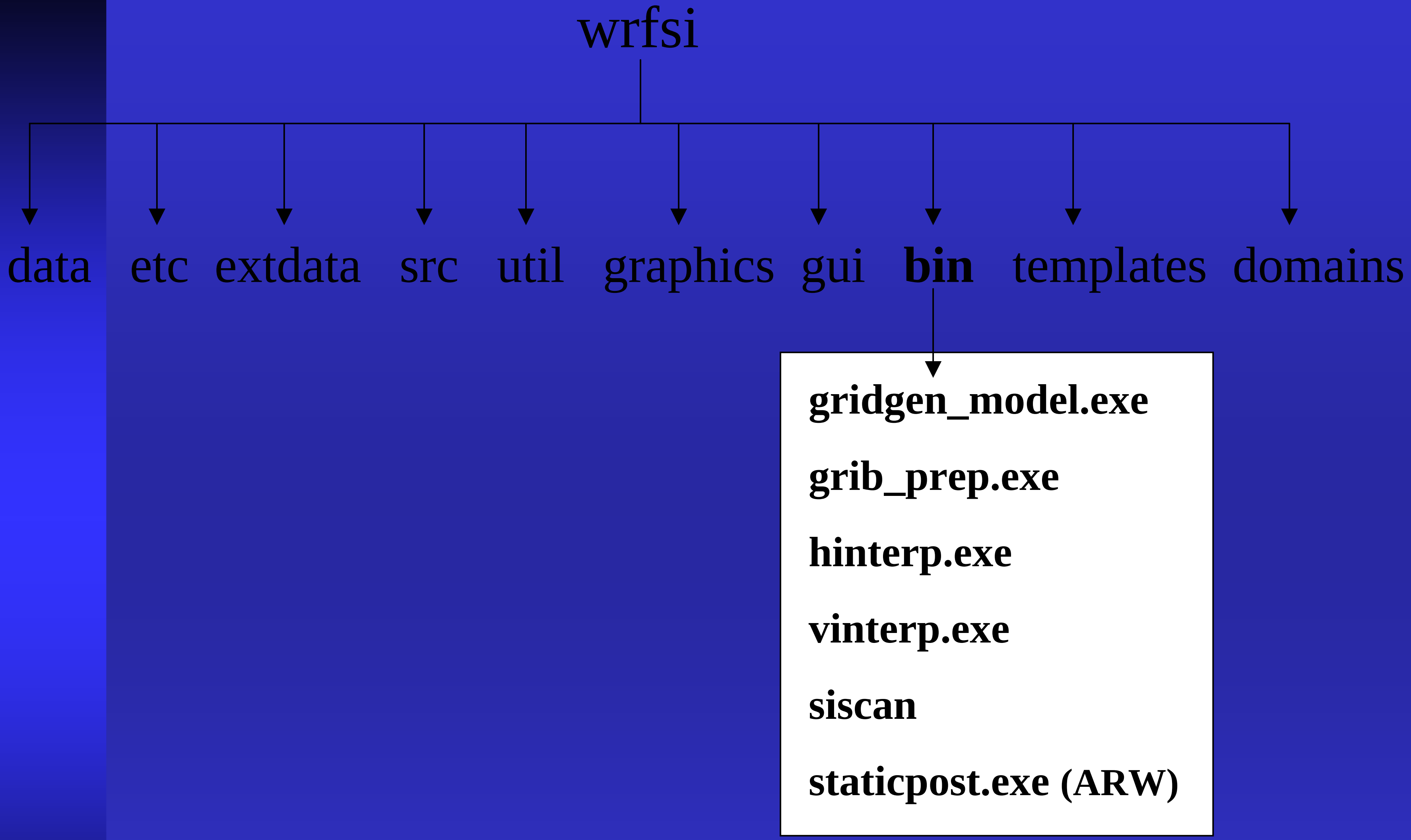
- See file INSTALLROOT/**config_paths**
    - Created by install_wrfsi.pl
    - Lists all the environment variable paths
    - Can be used in a later session to redefine SI env paths, especially for use with the GUI

Installing the Software

5. Check for a successful installation

   - Status information will be written to the screen during the build with additional information written to a log file SOURCE_ROOT/make_install.log.

   - Check INSTALLROOT/bin for the executables:
     - gridgen_model.exe
     - grib_prep.exe
     - hinterp.exe
     - vinterp.exe
     - siscan, and
     - staticpost.exe (ARW)
   - Check INSTALLROOT/etc for scripts:
     - window_domain_rt.pl
     - grib_prep.pl
     - wrfprep.pl



wrfsi

data etc extdata src util graphics gui **bin** templates domains

gridgen_model.exe

grib_prep.exe

hinterp.exe

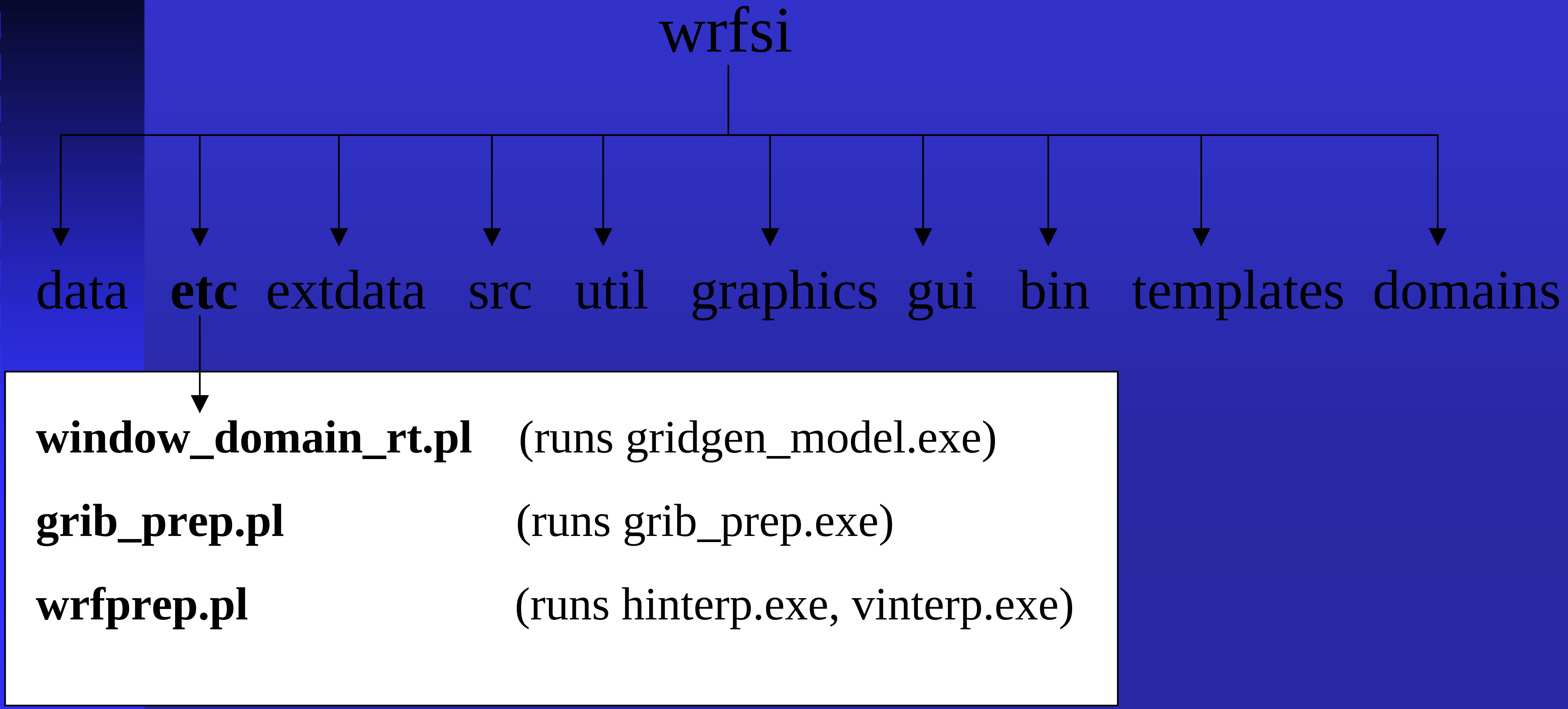
vinterp.exe

siscan

staticpost.exe (ARW)

Check for Fortran executables

to indicate a successful wrfsi installation after running install_wrfsi.pl.



wrfsi

data etc extdata src util graphics gui bin templates domains

window_domain_rt.pl (runs gridgen_model.exe)

grib_prep.pl (runs grib_prep.exe)

wrfprep.pl (runs hinterp.exe, vinterp.exe)

Check for Perl scripts

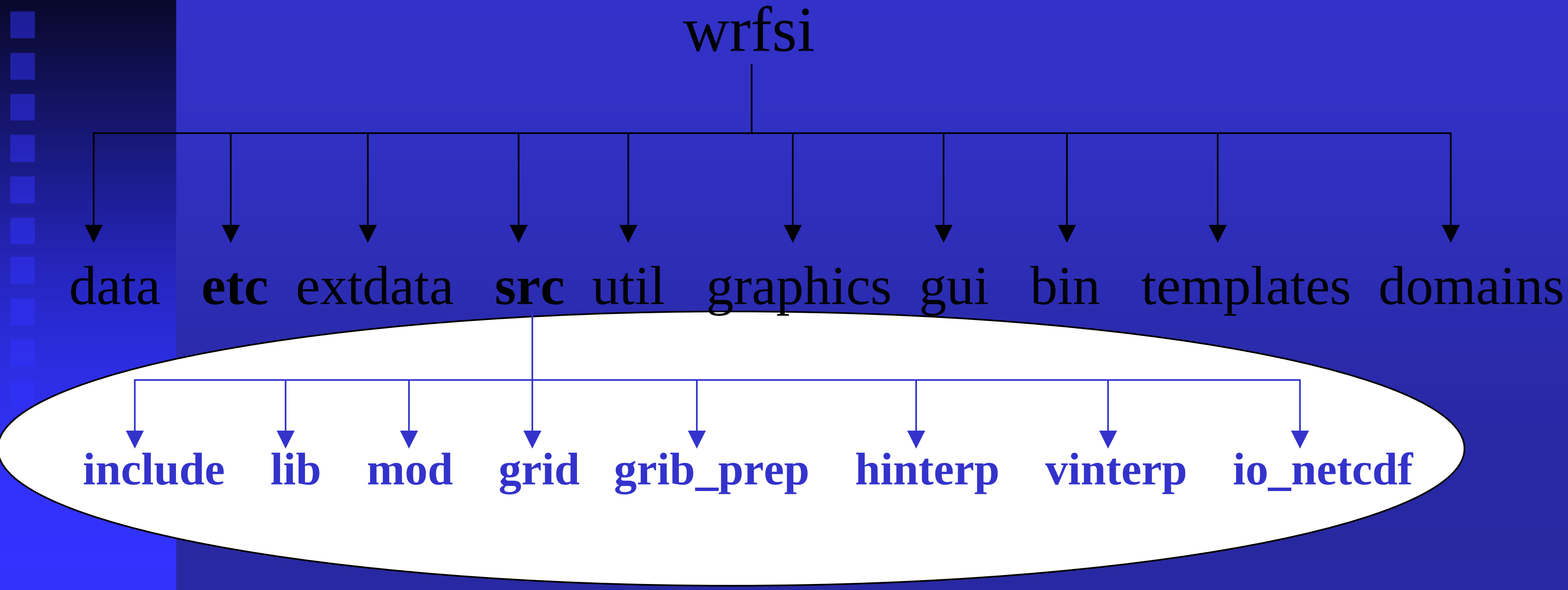
to indicate a successful wrfsi installation after running install_wrfsi.pl.

Installing the Software

5. Check for a successful installation (con't)

- If an executable is missing from INSTALLROOT/bin
  - cd $SOURCE_ROOT/src
  - Look for the subdirectory used to build each executable
  - cd to the corresponding directory

    (e.g. cd grid for gridgen_model.exe)
  - Run 'make' to manually build the desired executable
  - On success, run 'make install' to move the exe to INSTALLROOT/bin.
  - If compile errors are found try to debug the software for your system, or ask for help from wrfhelp@ucar.edu.



wrfsi

data **etc** extdata **src** util graphics gui bin templates domains

include lib mod grid grib_prep hinterp vinterp io_netcdf

Look for src directory

used to build each executable in the event that an executable is not found in INSTALLROOT/bin.

Installing the Software

5. Check for a successful installation (con’t)

   - If NetCDF is not built with the same complier used to build SI, you need to rebuild NetCDF
   - If you encounter and correct a software issue, feel free to then rebuild SI from the beginning if desired
   - Also, please forward any problems and their solutions, if possible, to wrfhelp@ucar.edu

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Configuring Domains

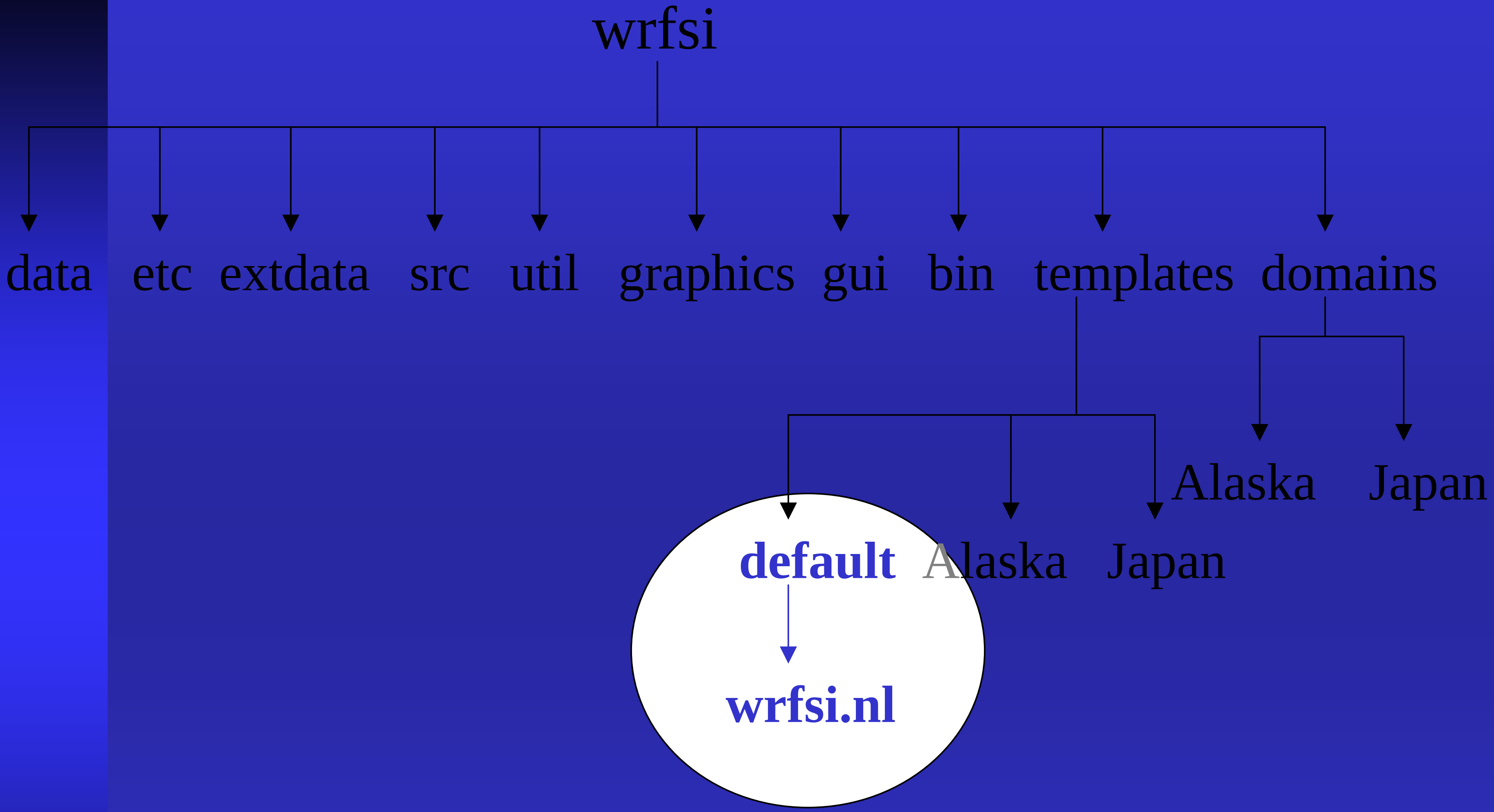
- Configuring a Domain Overview

  To Define and localize a domain

1. Create a template and domain directory for your domain
2. Copy the wrfsi Fortran namelist, wrfsi.nl, and edit to your liking, esp. the domain specifications
3. Run localization perl script – window_domain_rt.pl
4. Check for a successful localization

Configuring Domains

1. Create a template and domain directory for your domain
   - Each domain you create needs new “my-case” subdirectory under TEMPLATES and DATAROOT
     - mkdir $TEMPLATES/my-case
     - mkdir $DATAROOT/my-case



Principal copy of wrfsi.nl

After installation processes, location of the principal copy of the SI Fortran namelist, wrfsi.nl, will be found in TEMPLATES/default/wrfsi.nl.

Configuring Domains

2. Copy the wrfsi Fortran namelist, wrfsi.nl, and edit to your liking, esp. the domain specifications

   - cp TEMPLATES/default/wrfsi.nl to TEMPLATES/my-case
   - chmod –R u+w TEMPLATES/my_case
   - cd TEMPLATES/my-case

Configuring Domains

2. Copy the wrfsi Fortran namelist, wrfsi.nl, and edit (con’t)
   - Edit wrfsi.nl
     - &project_id section (simulation_name and user_desc)
     - &hgridspec section (sets up horizontal domain)
       - XDIM, YDIM = dimension in E-W and N-S direction, respectively
       - MAP_PROJ_NAME =
         - ARW – 'lambert', 'mercator', or 'polar‘
         - NMM – ‘rotlat’
       - MOAD_KNOWN_LAT/LON = center latitude (+ N), longitude (+ E)
       - MOAD_STAND_LATS = true latitude1 (lambert and polar), true latitude for lambert (set to +/- 90 for polar)
       - MOAD_STAND_LONS = orientation longitude
       - MOAD_DELTA_X and Y = grid spacing in the x and y directions
         - ARW – in meters
         - NMM – in radian degrees
   - We recommend removing all namelist sections and vars that do not differ from those found in TEMPLATES/default/wrfsi.nl

Configuring Domains

3. Run localization perl script – window_domain_rt.pl

   - Main task of script is to run gridgen_model
   - Check that your SI environment vars are configured
   - Run INSTALLROOT/etc/**window_domain_rt.pl**
     - Required flags:
       - -t ‘Directory path to domain subdirectory; for example, TEMPLATES/”my-case”’
       - -w wrfsi (ARW), or wrfsi.rotlat (NMM)
     - Optional flags:
       - -s, -i, and -d flags will override the environment variables if necessary
       - -c flag reconfigures the entire MOAD_DATAROOT; ie. removes directories log, siprd, static, and cdl. Without -c only the static and cdl directories are rewritten.
     - Example command line:

```
perl window_domain_rt.pl -w wrfsi -t $TEMPLATES/”my-case”
```

Configuring Domains

4. Check for a successful localization

- Running ‘window_domain_rt.pl’ will result in one of the following messages.

success:

“window_domain_rt complete”

failure:

“Lines with error found in localize_domain.log”

“→ localization incomplete ←”

“window_domain_rt incomplete”

- Helpful print statements are written from window_domain_rt.pl to MOAD_DATAROOT/log/**localize_domain.log** for more detailed information

Configuring Domains

4. Check for a successful localization (con’t)

- Common Error conditions:
  - Geog path is not set properly
  - Not enough geog tiles to cover your domain (especially true if you get only one or two of the quarter sphere geog tar files)
  - Perhaps other values in wrfsi.nl are not set properly. Note that “window_domain_rt.pl” will save and copy the static subdirectory (called MOAD_DATAROOT/**static_err**) in the event of an error for you to check

- The existence of static file indicates success:
  - ARW check for MOAD_DATAROOT/static/**static.wrfsi.d01**
  - NMM check for MOAD_DATAROOT/static/**static.wrfsi.rotlat**
  - Both are netCDF files that can be viewed with ncdump or ncBrowse utility

Configuring Domains

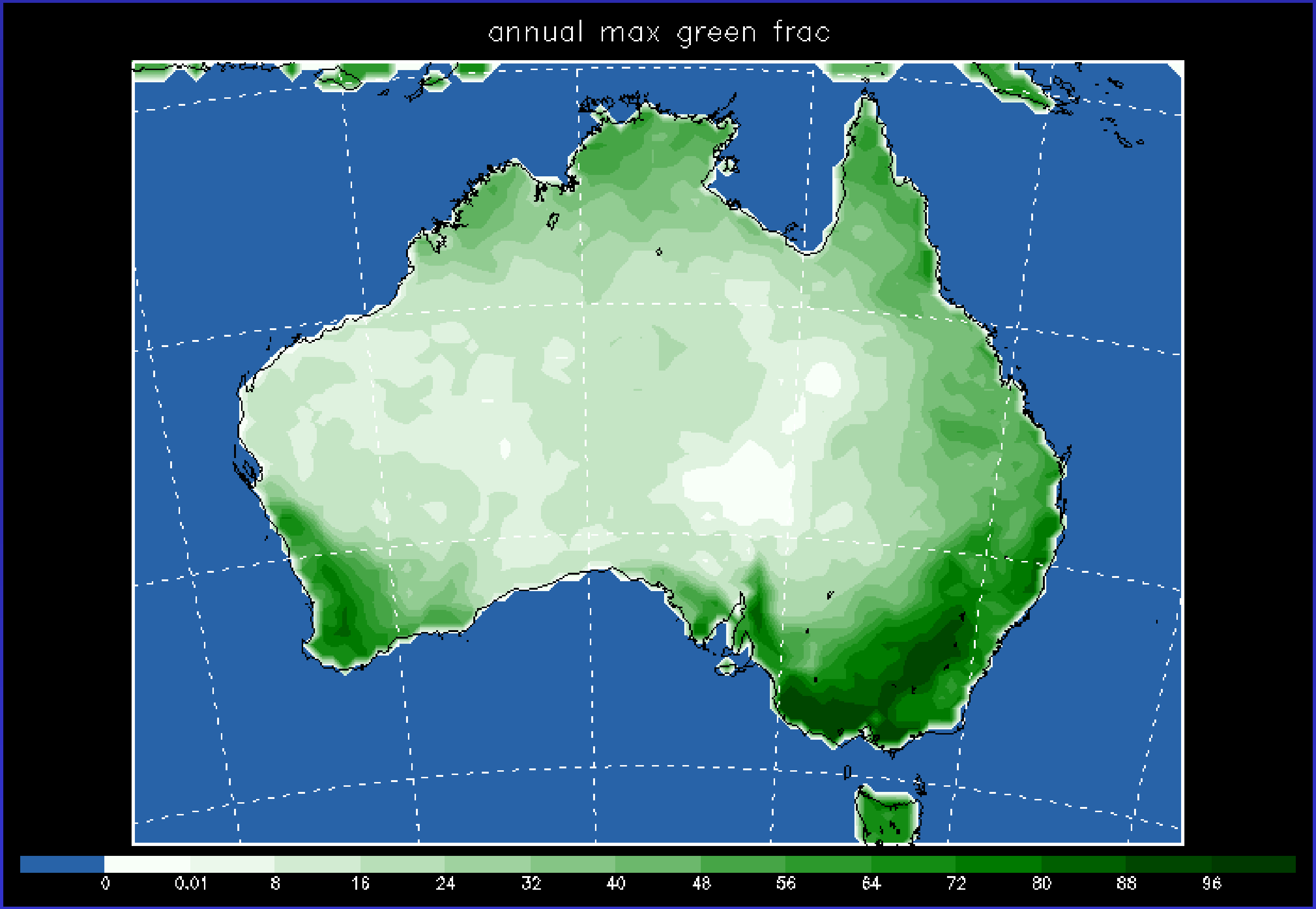
4. Check for a successful localization (con’t)

- To confirm domain localization generate graphical gmeta images
  - setenv NCARG_ROOT /usr/local/ncarg-4.3.0
  - setenv NCL_COMMAND $NCARG_ROOT/bin/ncl
    - Or, see wrfsi.noaa.gov/gui/faq_ncl

  - cd $INSTALLROOT/graphics/ncl
  - generate_images.pl -domain=/wrfsi/domains/Alaska
  - idt /wrfsi/domains/Alaska/static/meta.d01.ncgm

Configuring Domains

4. Check for a successful localization (con't)



Overview

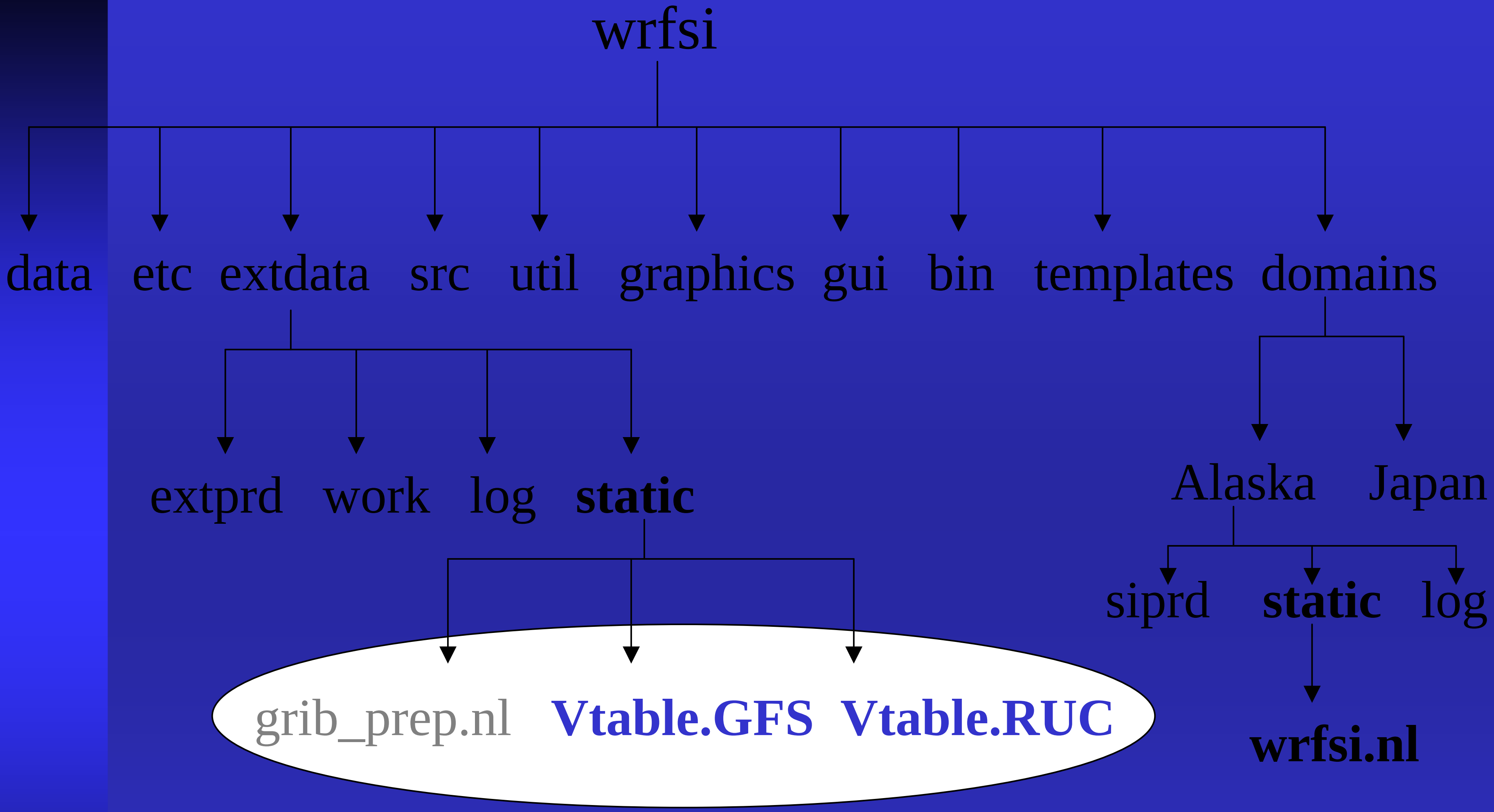
- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Configuring Interpolation

- Configuring Interpolation Overview
  - grib_prep Configuration
  - wrfprep Configuration

Configuring Interpolation

- grib_prep configuration
  - GRIB Data
    - Acquire time-varying meteorological data for WRF’s initial and lateral boundary conditions
  - GRIB Decoding
    - A Vtable (variable table) file contains information to extract model data from a GRIB file to initialize WRF
      - Vtable.GFS, Vtable.ETA, etc. are provided
      - Contains a list of variables and their levels based on standard GRIB codes; e.g. 11 is temp, 1 is surface level
      - Allows output name re-specification for each variable; e.g. above the parm name is skin temperature, TSK.
    - Confirm that there is a Vtable for your selected GRIB data file available in EXT_DATAROOT/static/Vtable.{SOURCE} where SOURCE is, e.g. GFS
    - Easy to create a Vtable.”newGRIBsource” (depending on model)



wrfsi

data etc extdata src util graphics gui bin templates domains

extprd work log **static**

grib_prep.nl **Vtable.GFS** **Vtable.RUC**

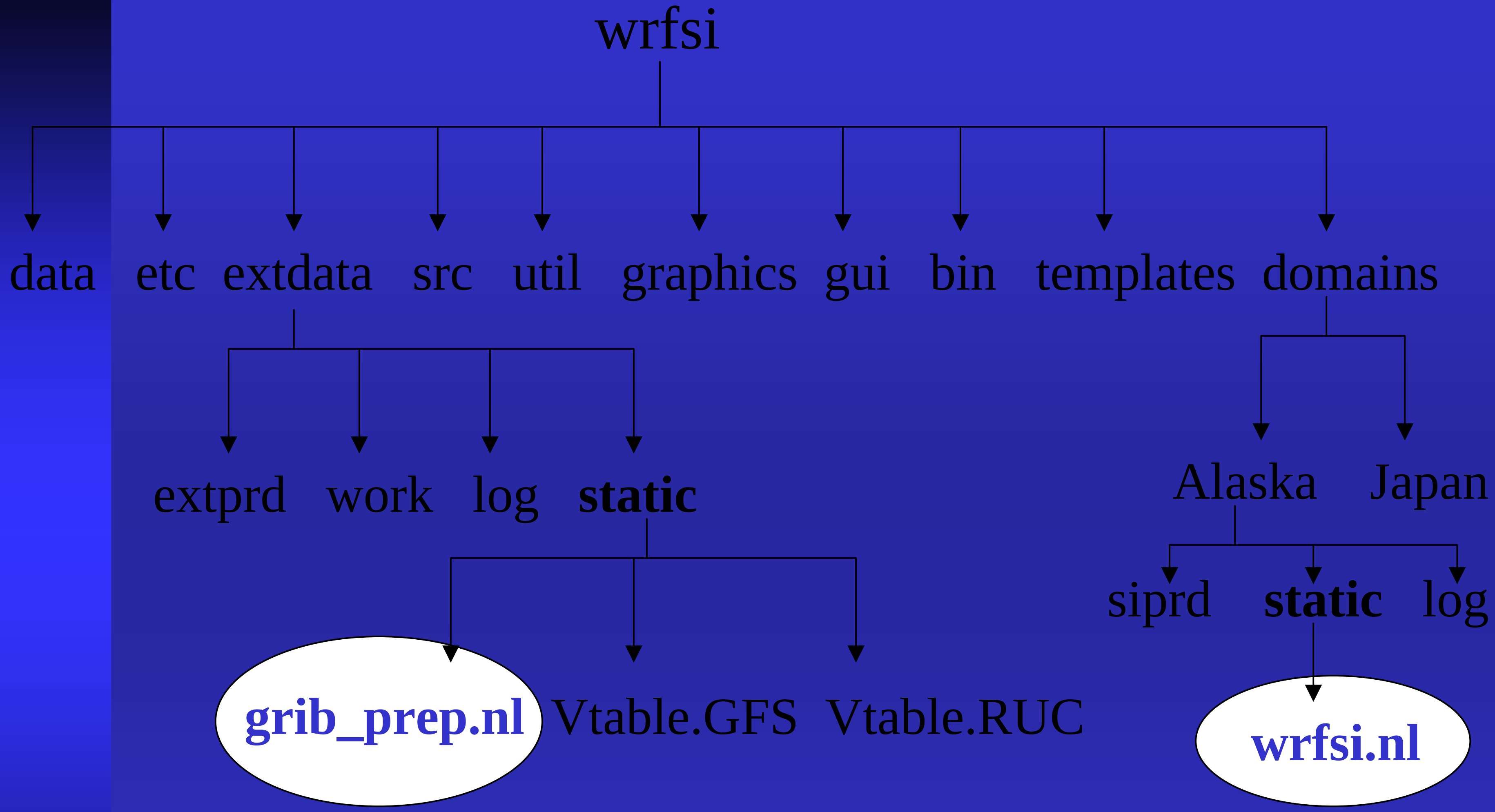
Alaska Japan

siprd **static** log

wrfsi.nl

GRIB Vtable

A variable table file containing information to extract model data from a GRIB file to initialize WRF.



Principal Fortran Namelists

Edit grib_prep.nl for **grib_prep.pl** to locate and decode GRIB files.
Edit wrfsi.nl for **wrfprep.pl** to locate and process your extracted GRIB data.

Configuring Interpolation

- grib_prep configuration (con’t)
    - Edit EXT_DATAROOT/static/grib_prep.nl
        - &filetimespec
            - Set start and stop times
            - For real-time runs these values would be set automatically
        - &gpinput_defs
            - Set frequency of GFS and ETA grid receipt time and typical time delay, in hours, after cycle time, as well as Vtable name

```
SRCNAME = 'GFS', ‘ETA‘
SRCVTAB = 'GFS', ‘ETA’
SRCPATH ='/public/data/grids/gfs/0p5deg/grib',
                '/public/data/grids/40km_eta/grib‘
SRCCYCLE = 3, 3
SRCDELAY = 1, 0
```

Configuring Interpolation

- wrfprep configuration
  - Edit DATAROOT/my-case/wrfsi.nl (or, if prior to running window_domain_rt.pl edit TEMPLATES/my-case/wrfsi.nl)
  - &interp_control
    - PTOP_PA: Top pressure level to consider from input data
    - HINTERP_METHOD and LSM_HINTERP_METHOD
      - 0 = nearest neighbor (not recommended for HINTERP_METHOD)
      - 1 = 4-point linear
      - 2 = 16-point quadratic
    - INIT_ROOT,
    - LBC_ROOT, and
    - LSM_ROOT: Model source prefixes (e.g. ETA:) of binary data files in EXT_DATAROOT/extprd to use for dynamic data

      (E.g. to run with ETA model and SST for LSM set
      - INIT_ROOT='ETA', LBC_ROOT='ETA' and LSM_ROOT='SST')

Configuring Interpolation

- wrfprep configuration (con't)
  - CONSTANTS_FULL_NAME: Full file names of data located in EXT_DATAROOT/extprd that contain values to be held constant for entire run
  - VERBOSE_LOG: Set to true for more extensive logging
  - LEVELS: List of “full” vertical levels from bottom to top of atmosphere
    - ARW: starting at 0 for “ZETA” and starting at 1.0 for “ETAP”
    - NMM: starting at 1.0 for “NMMH”
  - OUTPUT_COORD: which vertical coordinate to use.
    - ARW
      - “ZETA” specifies the height-based R-K WRF or
      - “ETAP” specifies the mass version.
    - NMM
      - “NMMH” specifies the hybrid vert. coordinate
  - Its recommended to leave all other settings as they are

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Running

- Running the SI Overview
  - Confirm set up
  - Run grib_prep.pl – for GRIB data pre-processing
  - Run wrfprep.pl – for horizontal, vertical interpolation, and grid staggering
  - (Or, optionally run wrfsi.pl – to run both grib_prep.pl and wrfprep.pl)
  - (What happens when script wrfsi.pl runs)
  - Did code run properly
  - Check the output

Running

- Confirm set up
  - Ensure GFS or ETA GRIB data is available in the specified directory, for requested time intervals
  - Note recognized GRIB file naming conventions are:
    - FSL: yyjjjhhmmffff
    - NCEP: ???.ThhZ.??????ff.????
    - NCEP: ???_yymmdd_hh_ff
    - And others
    - If your files do not conform to these naming conventions, use “-f” with the first several characters of the filename (-f avn2004)
    - Otherwise, the grib_prep script will assume every file is a possible match
  - Ensure domain configuration step has been successfully completed
  - Ensure wrfsi.nl has the correct EXT_DATAROOT path value
  - Are INSTALLROOT and MOAD_DATAROOT path values set

Running

- Run grib_prep.pl
  - grib_prep.pl *manages* the time-series of data to process
  - grib_prep.pl *runs* the Fortran executable grib_prep.exe – which loads the Fortran namelist EXT_DATAROOT/static/grib_prep.nl setting all the necessary filenames and values
  - grib_prep.exe *decodes, extracts* and *time interpolates* GRIB model parameters

  - Output is written to EXT_DATAROOT/extprd
  - Log information is written to EXT_DATAROOT/log
  - While files are being created they are written to EXT_DATAROOT/work/GFS (or other source name)

  - Run

```
$INSTALLROOT/etc/grib_prep.pl -s 200508241200 -l 12 -t 6 GFS
```

or $INSTALLROOT/etc/grib_prep.pl -help (for command options)

Running

- Run wrfprep.pl
  - wrfprep.pl *uses* values set in the Fortran namelist wrfsi.nl section &interp_controls to locate and process the data
  - wrfprep.pl *runs* the Fortran executables: hinterp.exe and vinterp.exe
  - hinterp.exe and vinterp.exe *interpolate* the previously extracted GRIB model parameters to fit the user defined domain.

  - Interpolated output files are written to MOAD_DATAROOT/siprd
  - Log files are written to MOAD_DATAROOT/log for the hinterp and vinterp runs.

  - Run

```
$INSTALLROOT/etc/wrfprep.pl -s 200508241200 -f 12
```

or $INSTALLROOT/etc/wrfprep.pl -help (for command options)

Running

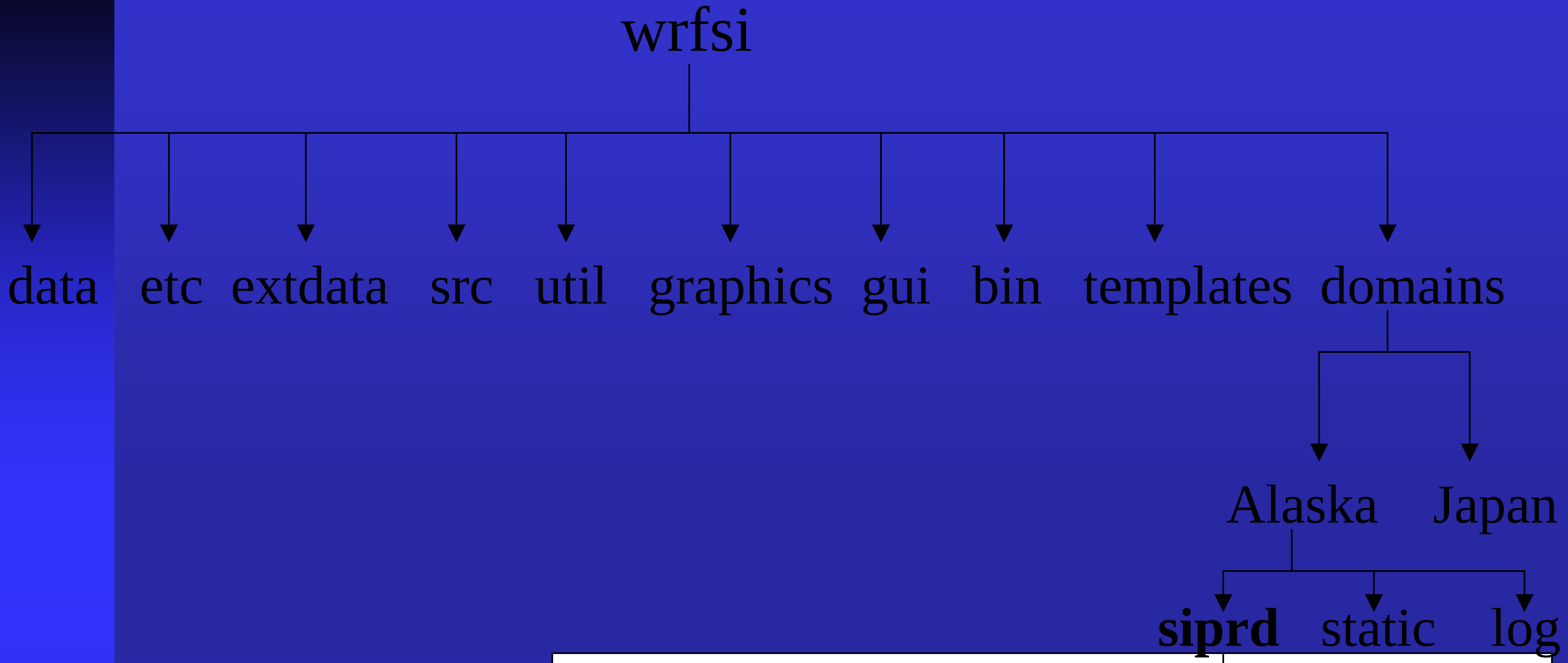
- (Or, optionally run wrfsi.pl – to run both grib_prep.pl and wrfprep.pl)
    - Run script with 4 arguments:
    - YYYYMMDDHH: Year, month, day, and hour UTC of model start time
    - FF: Length of forecast to be produce in hours
    - Source: GFS or ETA
    - Domain Name: A name for this run, my-case
    - Example:
        - $INSTALLROOT/etc/wrfsi.pl 2005082412 24 ETA myrun
    - You can optionally provide the dataroot and installroot using command line options -d and -i.
        - wrfsi.pl -d /my/dataroot -i /my/installroot 2005082412 24 ETA my-case

Running

- (What happens when script wrfsi.pl runs?)
  - This script edits MOAD_DATAROOT/static/wrfsi.nl for run time initialization and model source
  - This script calls grib_prep.pl which runs grib_prep.exe
  - This script calls wrfprep.pl which runs hinterp.exe and vinterp.exe
  - Output from script is written to MOAD_DATAROOT/siprd
  - A log file is written to EXT_DATAROOT/log for the grib_prep run
  - Log files are written to MOAD_DATAROOT/log for the hinterp and vinterp runs.

Running

- Did code run properly?
  - Check for output in MOAD_DATAROOT/**siprd**
    - From previous run example, we should have
    - For ARW:
      - wrf_input.global.metadata
      - wrf_input.d01.2005-08-24-12:00:00
      - wrf_input.d01.2005-08-24-15:00:00
      - ...
      - wrf_input.d01.2005-08-25-12:00:00
    - For NMM:
      - wrf_input_nm.*
    - Frequency of output files is set in &filetimespec portion of wrfsi.nl
  - If correct files are not present, check the diagnostic log files for information to determine what is needed.
    - Consider rerunning wrfprep.pl with e.g. -o (offset time) -2 (previous 2 hrs)



wrfsi

data etc extdata src util graphics gui bin templates domains

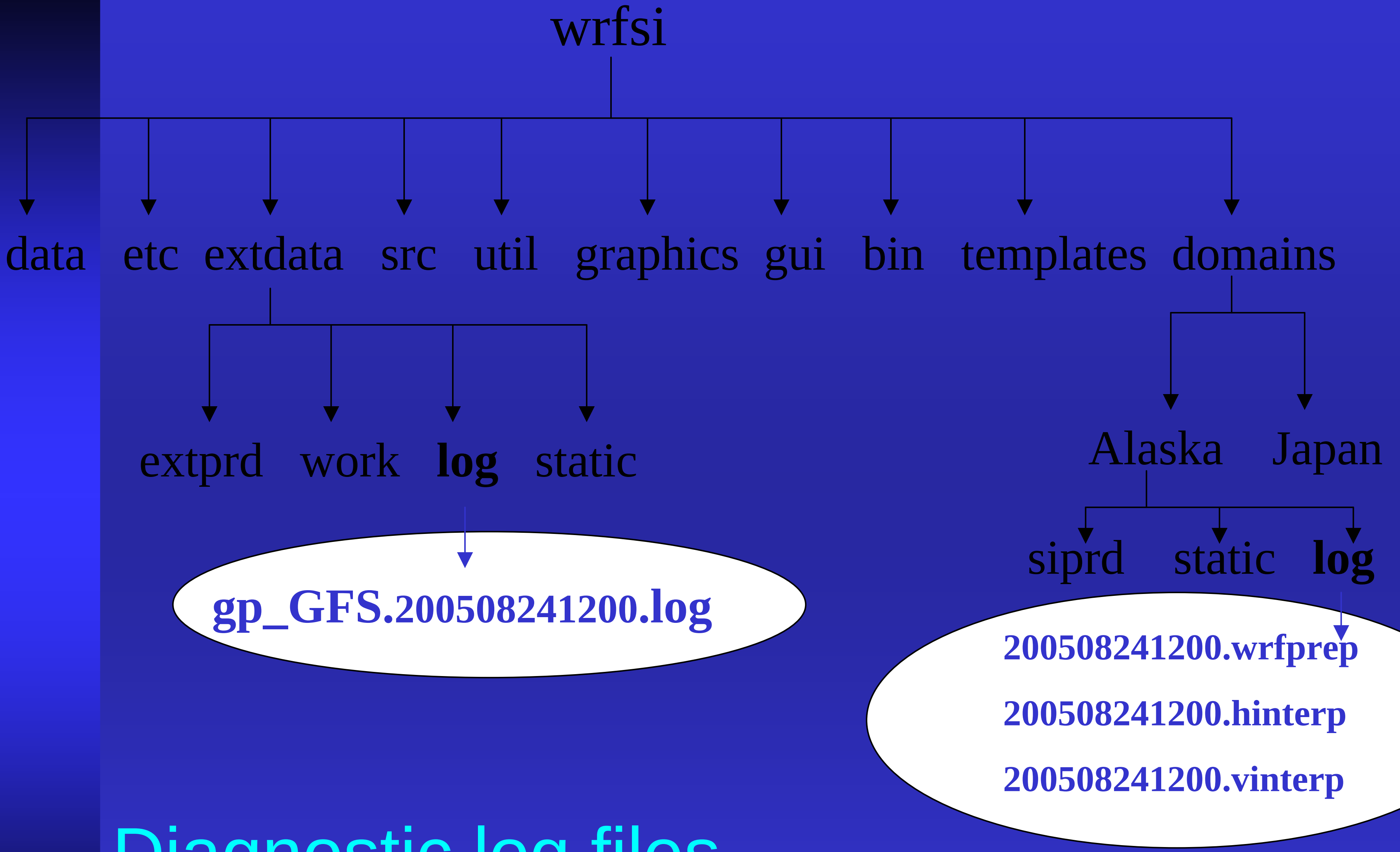
Alaska Japan

siprd static log

- **wrf_input.global.metadata**
- **wrf_input.d01.2005-08-24_12:00:00**
- **wrf_input.d01.2005-08-24_18:00:00**
- **wrf_input_nm.d01.2005-08-24_18:00:00 (NMM)**

Look for input to WRF model

to indicate successfully running hinterp and vinterp, via wrfprep.pl.



wrfsi

data etc extdata src util graphics gui bin templates domains

extprd work **log** static

gp_GFS.200508241200.log

Alaska Japan

siprd static **log**

200508241200.wrfprep

200508241200.hinterp

200508241200.vinterp

Diagnostic log files

Evaluate grib_prep’s gp_* log file to help determine what was needed for GRIB data.
Evaluate wrfprep’s log files to generate WRF input data.

Running

- Checking the output
  - Use the INSTALLROOT/bin/siscan program to dump a summary of file contents
    - siscan {file}
  - ARW: IDL routines in SOURCE_ROOT/util read the hinterp and vinterp output files.
- Other Notes
  - Executables can be run directly without the use of any scripts by simply setting the MOAD_DATAROOT environment var and ensuring the namelists are correct
  - Consider using the WRFSI GUI to accomplish all of the SI processes.
  - See wrfsi.noaa.gov for additional SI and GUI information

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Initializing the WRF Model

- Edit the WRF *namelist.input* file to make it consistent with the WRFSI domain configuration
  - ztop, dx, dy, io_form, etc.

- Run the WRF real routine using the WRFSI output files
  - ARW: Run real.exe using the wrf_input.* as input
  - NMM: Run real_nmm.exe using wrf_input_nmm.*

- You can then run the model!

- See Dave Gill's presentation for more ARW details, and Matt Pyle’s presentation for more NMM details.

Overview

- SI Capabilities
- Source Code
- System Requirements
- Installing the Software
- Configuring Domains
- Configuring Interpolation
- Running
- Initializing the WRF Model
- Summary

Summary

- Setting up and running the WRFSI is done by:
- 1) Building the software
- 2) Configuring your domain
  - Domain configuration is easy with the use of templates and localization scripts
- 3) Running the grib_prep.pl and wrfprep.pl scripts
- The SI system is flexible enough to run each component separately

- The WRFSI development at FSL is complete and operational.
  - There are plans to unify the SI package for the two dynamic cores. This effort is being considered by NCAR, who is further developing the SI software to optimize and parallelize several components of the package.
  - Bug fixes and minor enhancements will be done as resources permit
  - We welcome feedback, bug reports, etc.